CEWES Major Shared Resource Center

MSRC

CEWES MSRC/PET TR/99-28

Simulation Code Launching From the Web

by

Chandrajit Bajaj
Steven Cutchin
Mary Wheeler

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC94-96-C0002
Nichols Research Corporation

SIMULATION CODE LAUNCHING FROM THE WEB

Chandrajit Bajaj and Steven Cutchin and Mary Wheeler
Department of Computer Sciences and TICAM
University of Texas, Austin, TX 78712
http://www.ticam.utexas.edu/CCV

1 Accomplishments

Focussed effort two of the PET/CEWES project has been to develop web-based tools which serve as prototypes for launching parallel simulations from remote environments. The parallel simulations of interest arise in subsurface flow and transport, but the tools to be developed will be of general use. PARSSIM (Parallel Aquifer and Reservoir Simulator), a parallel, three-dimensional, flow and transport simulator for modeling contamination and remediation of soils and aquifers is being used in this demonstration. The code was developed at the University of Texas and contains many of the features of current state-of-the-art groundwater codes. It is fully parallelized using domain decomposition and MPI and is operational on the IBM-SP and CRAY T3E platforms.

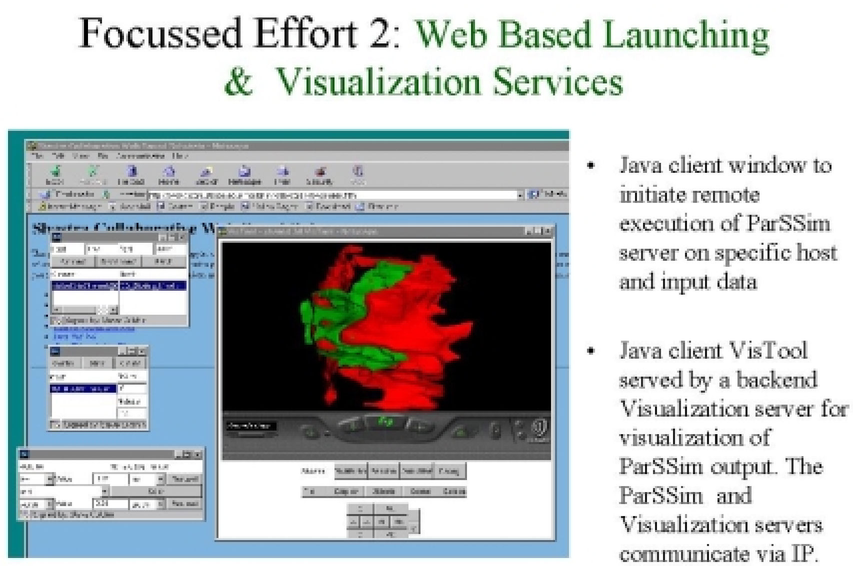


FIGURE 1: The web based visualization and steering applet displaying iso-surfaces from a PARSSIM simulation.

A client java applet with GUI (graphical user interface) has been developed that allows remote users to access the PARSSIM code and data domains on CEWES MSRC servers. The results of the computation are then saved on the CEWES MSRC local disks and also selectively sent back to the requesting java applet. A snapshot of the java applet can be seen in Figure 1. The java applet can be instantiated from any internet web browser. As a first prototype of launching, we have created tools appropriate for executing PARSSIM on

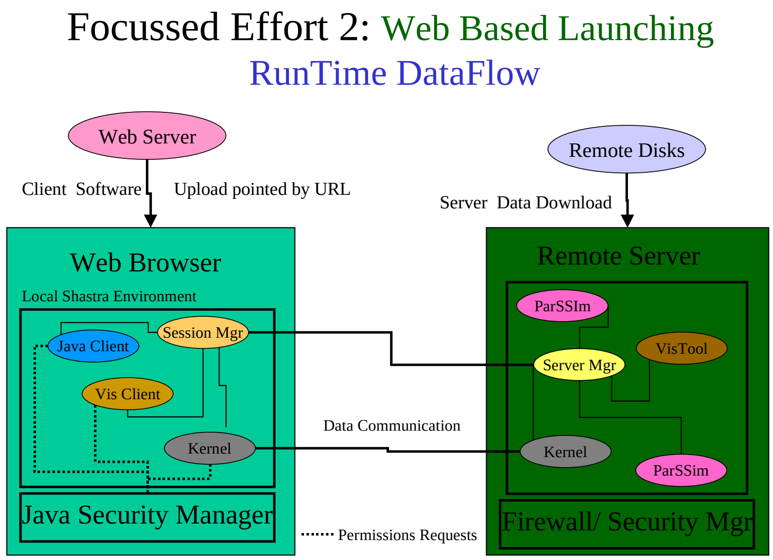


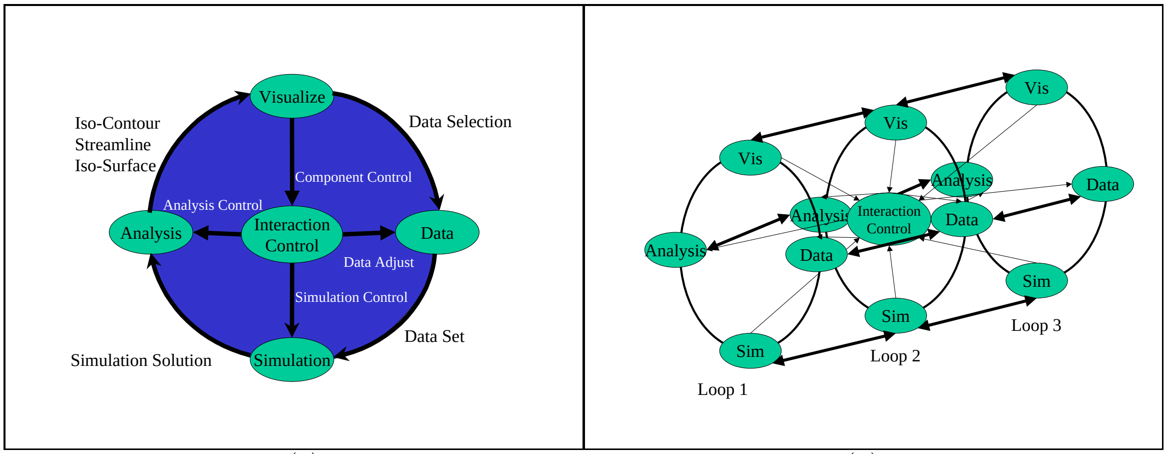
FIGURE 2: An overview of the visualization clients, webserver, visualization server the PARSSIM servers simulation architecture and interprocess communication.

a single computing environment. We are now investigating the use of more complex programming tools such as CORBA, and GLOBUS, which provide the tools needed to execute in a metacomputing environment. A diagram showing the web interaction of the web applet and remote servers is shown in Figure 2.

2 Software Architecture Details

Scientific visualization involves a collection of data sets, a simulation system, analysis packages and visualization software. Data sets can be quite large easily involving gigabytes of information. Simulators can take long times to execute on the order of weeks. Analysis packages cull the important information from the results of simulations and data sets to provide useful information for the visualization software. Finally the visualization software displays the results of the analysis for the visualizer to view. Using most current systems often this whole process must be re-run to generate new visualization from the same data sets, although significant steps forward have been made in introducing more interactivity into the process. Figure 3(a) presents this traditional visualization scheme. In order to support distributed scientific visualization we extend this loop model of scientific visualization into the third dimension. Figure 3(b) depicts this extension to the loop model. A single collection of data source, simulator, analysis tools, and a visualization client makes up a data-simulation-analysis-visualization(DSAV) loop. A DSAV loop is the basic unit of model organization for work. Users can selectively add and remove components from a loop. Users own the loops that they create and only the owner of a loop may add and remove component members. All data generated by a component is owned by the loop the component belongs too. Users (including non-owners) may specify connections between components in different loops. The DSAV loop and cylinder model gives users a mental picture of the connection of distributed components and the work being done. Loops create in the users mind a notion of a common project. Connections create notions of a common task. Inter-loop connections specify those tasks that will assist in the completion of the loop project. Intra-loop connections specify those tasks that will assist in the completion of an external project from the components DSAV loop.

This new interaction model has many benefits for distributed visualization of simulations. Use of the



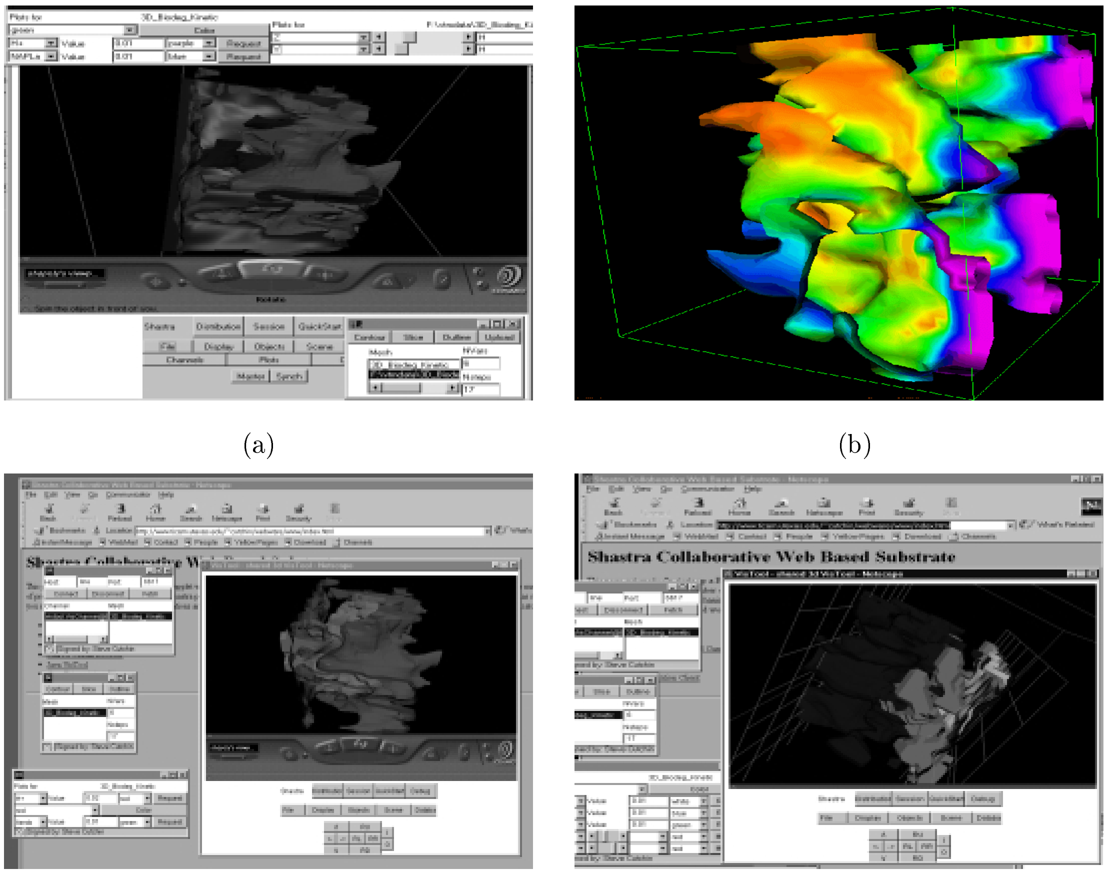
(a) (b)

Figure 3: A data, simulation, analysis, visualization (DSAV) loop which displays the communication flow between components of a simulation visualization project is shown in (a). (b) displays the collaborative version of the SAV loop. It depicts the multi-loop cross-loop nature of collaborative interaction.

model changes the nature of the work to the distributed manipulation of simulation, analysis and visualization tasks without regard to user location or schedule. It also provides a framework for easily distributing tasks over a larger pool of network resources, increasing the available resources in terms of both computers, software and personnel. The new model also creates an environment that supports shared task management and provides collaborative task awareness. The use of the Shastra substrate in conjunction with the model provides flexible coupling to support user determined division of labor for working on simulation visualization and control. The model also enhances the ability to arrange the organization of multiple views of the same data set. The model also supports a more loosely coupled manipulation of resources and tasks than other strictly hierarchicaly models.

Our application demonstrating the utility of the DSAV loop interaction model is the visualization of hydraulic conductivity in soil simulation. Figure 4 shows sample screenshots of the visualizations of the results of the simulation. The user interacts with the various components of the DSAV loop through the visualization applet. The visualization applet provides the user-interface for creating shareable DSAV loop's. Within the visualizer the user can select and specify flow solvers and vistool servers to include within specific DSAV loops. A flow solver, vistool server, and data set may belong to only one DSAV loop. It may however be connected to other DSAV loops if the DSAV loop it belongs to allows this. Multiple DSAV loops may be created as needed by visualization users. Every component of the DSAV loop has a unique network id. Input and output data sets for DSAV components also have unique network id's and are associated with the DSAV loops that created them. This network id can be permanently associated with the data set if the user wants to. This supports asynchronous manipulation of shared data. Users may selectively start and stop components on the workstations that they have access to.

The PARSSIM flow solver may be controlled in a similar manner to vistool, data sets with operations may be explicitly submitted and retrieved, or connections may be placed between the solver and vistool, or the visualizer with specific operations specified on the connections. PARSSIM takes a finite element mesh representation of a fluid flow problem and produces a mesh representation with the flow values stored at the vertices of the mesh. This result may then be distributed to analysis and visualization tools.



(a) (b)

(c) (d)

FIGURE 4: (The surface in these images are an isocontour of the hydraulic conductivity of the soil. The surface is colored according to the concentration of a contaminant (green, low; magenta, high) dissolved in the groundwater. Picture (a) presents the visualization within the web based applet. Picture (b) is a close up of the surface visualized. Pictures (c) and (d) present two more visualizations within the visualization applet.